

Pandora Analysis Calibration Executables

Steve Green, sg568@hep.phy.cam.ac.uk

Abstract

The calibration procedure required for simulation and reconstruction of high energy physics events at a future electron positron collider using the particle flow algorithm PandoraPFA is described in full. The calibration procedure addresses how to set the constants used in the simulation for both digitisation, the estimation of energy deposits in the absorber material of the sampling calorimeters based on the energy deposits in the active material, and those used within PandoraPFA to set the hadronic and electromagnetic energy scales for the various particle showers being reconstructed. To be assured of accurate reconstruction, recalibration must be performed for all changes to detector design and/or reconstruction logic in PandoraPFA.

These studies were developed in the context of simulation of ILD (ILD_o1_v06) using Mokka for the detector simulation. This procedure can be adapted to other contexts as fundamentally the logic applied is universal to simulation of sampling calorimeters. Please contact the Cambridge linear collider group for any assistance.

1 Introduction

To ensure accurate results from the simulation of the high energy physics events at a future electron positron collider it is necessary to accurately define a number of constants used in both the simulation and reconstruction. The calibration procedure described here addresses the setting of the calibration constants used in both the digitisation processor (ILDCaloDigi or NewLDCCaloDigi) and PandoraPFA.

The purpose of the constants used in the digitisation processor is twofold: (i) To set a series of constants that accurately estimate the amount of energy deposited in the absorber material of the sampling calorimeter being simulated based on the measured energy desposits in the active material and (ii) to set the MIP scale used in the digitisers.

It should be noted that the MIP scale only needs setting in the digitisers if you wish to apply the realistic digitisation options in ILDCaloDigi. In such a case the MIP scale is needed to apply a dynamic range to the readout electronics in the calorimeters.

The purpose of the constants used within the particle flow algorithm PandoraPFA is to set the different energy scales for electromagnetic and hadronic particle showers. There are a total of four constants applied, which set the electromagnetic and hadronic energy scales in both the electromagnetic calorimeter (ECal) and the hadronic calorimeter (HCal). The

constants need not be the same for both calorimeters due to the different atomic structures of the calorimeters.

A detailed list of the calibration constants being defined in this procedure for the digitiser and for PandoraPFA can be found in table 1 and table 2 respectively.

Table 1: Digitisation Constants

Constant	Description
CalibrECAL	Converts the SimCalorimeterHit energy deposit (active material) into a CalorimeterHit energy deposition (active material and absorber material) in units of GeV in the ECal
CalibECALMIP	Sets the MIP energy scale in the ECal (for digitisation)
CalibrHCALBarrel	Converts the SimCalorimeterHit energy deposit (active material) into a CalorimeterHit energy deposition (active material and absorber material) in units of GeV in the HCal barrel
CalibrHCALEndcap	Converts the SimCalorimeterHit energy deposit (active material) into a CalorimeterHit energy deposition (active material and absorber material) in units of GeV in the HCal endcap
CalibrHCALOther	Converts the SimCalorimeterHit energy deposit (active material) into a CalorimeterHit energy deposition (active material and absorber material) in units of GeV in the HCal other (i.e. ring)
CalibHCALMIP	Sets the MIP energy scale in the HCal (for digitisation)

A series of executable files have been produced to determine the values that each of these calibration constants can take. These executables can be found within the calibration folder inside the PandoraAnalysis toolkit. A description on how to use each of these executables follows. It should be noted that the setting of these calibration constants is an iterative process.

2 Calibration Files

These executables calibrate the simulation and reconstruction using single particle events of known particle type and fixed particle energy. Therefore, before performing any analysis with the sector model you have to simulate and reconstruct photons, muons and long-lived neutral kaons. After running PandoraPFA in the Marlin framework, the MarlinPandora processor, on these single particle events, you must run the PfoAnalysis processor and save the output root files. These are to be used as inputs for the calibration executables.

To collect the relevant calibration information within the PfoAnalysis processor a series of extra parameters must be specified in the xml steering file in comparison to the

Table 2: PandoraPFA Constants

Constant	Description
ECalGeVToMIP	Converts SimCalorimeterHit energy deposits to units of number of MIPs in the ECal
HCalGeVToMIP	Converts SimCalorimeterHit energy measurements to units of number of MIPs in the HCal
MuonGeVToMIP	Converts SimCalorimeterHit energy measurements to units of number of MIPs in the Muon Chamber
ECalToEMGeVCalibration	Energy rescaling factor for electromagnetic energy in the ECal
HCalToEMGeVCalibration	Energy rescaling factor for electromagnetic energy in the HCal
ECalToHadGeVCalibration	Energy rescaling factor for hadronic energy in the ECal
HCalToHadGeVCalibration	Energy rescaling factor for hadronic energy in the HCal

default PandoraAnalysis setup. These include the sim calo hit collections produced from GEANT (via Mokka) and the calorimeter hit collections produced from the digitisation stage. You must also specify the CollectCalibrationDetails flag and some extra missing geometry information from the gear file. An example Marlin steering file set up for collecting all calibration information using this processor is shown in PandoraAnalysis/scripts/PandoraAnalysisCalibration.xml.

An example steering file that is set up to run the PfoAnalysis processor and collect the calibration information referenced above can be found in scripts/PandoraAnalysisCalibration.xml.

For the default ILD sector model it is recommended that you use 10 GeV photon samples, 10 GeV muon samples (this is a MIP definition and should not be varied) and 20 GeV long-lived neutral kaon samples.

The increase in energy for the kaon samples is suggested as these samples are used in general for calibrating parameters relating to the HCal and we ensure that, for a typical event, the bulk of the energy is deposited in the HCal and not in the ECal. Of course there will be some energy deposited in the ECal for certain events, but selection cuts on events used in calibration executables accounts for any effects that this may have on the calibration.

3 Digitisation

3.1 ECalDigitisation_ContainedEvents

This script is run on the root file outputs for the single photon events and determines how to set CalibrECAL.

Inputs:

1. Input root files, wildcards can be used.
2. True energy of photons being simulated.
3. Fractional calibration accuracy of digitisation step. This is used to define the bin width in the plot, which determines precision of results. Default: 2%
4. Output path to send the calibration text file and the output plot to.
5. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.

After running this script on the root files you rescale CalibrECAL by:

$$\frac{\textit{True Photon Energy}}{\textit{Mean Reconstructed Calo Hit Energy}} \quad (1)$$

Iteration is now performed until the total calorimeter hit energy for photon events contained in the ECal (produced by the script) falls within the desired accuracy. Iteration involves changing the CalibrECAL value in your Marlin steering files and rerun the photons through Marlin. This process is repeated until the total calorimeter hit energy for photon events contained in the ECal is within acceptable limits defined by the fractional accuracy of the digitisation step.

3.2 HCalDigitisation_ContainedEvents

This script is run on the root file outputs for the single long-lived neutral kaon events and aims to set CalibrHCALEndcap and CalibrHCALBarrel.

Inputs:

1. Input root files, wildcards can be used.
2. True energy of long-lived neutral kaon being simulated.
3. Fractional calibration accuracy of digitisation step. This is used to define the bin width in the plot, which determines precision of results. Default: 2%
4. Output path to send the calibration text file and the output plot to.

5. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.
6. Number of HCal layers in detector being simulated. Important for defining confined events.
7. Detector component string. Used as a naming convention. Options are either "Barrel" or "EndCap". This parameter is used as you perform separate digitisation for the Barrel and EndCap.
8. Lower CosTheta Cut. This is the lower limit of cos theta (polar angle), which you are using to define the Barrel or EndCap. Default: 0.2 Barrel, 0.8 EndCap
9. Upper CosTheta Cut. This is the upper value of cos theta (polar angle), which you are using to define the Barrel or EndCap. Default: 0.6 Barrel, 0.9 EndCap

After running this script on the kaonL root files you independently rescale CalibrHCAL-Barrel and CalibrHCALEncap by:

$$\frac{\textit{True KaonL Energy}}{\textit{Mean Reconstructed Calo Hit Energy}} \quad (2)$$

Iteration is now performed until the total calorimeter hit energy for long-lived neutral kaon events contained in the HCal Barrel and HCal EndCap (produced by the script) falls within the desired accuracy. Iteration involves changing the CalibrHCALEndcap and CalibrHCALBarrel values in your Marlin steering files and rerun the kaonL through Marlin. This process is repeated until the the total calorimeter hit energy for long-lived neutral kaon events contained in the HCal Barrel or HCal EndCap are within acceptable limits defined by the fractional accuracy of the digitisation step.

3.3 HCalDigitisation_DirectionCorrectionDistribution

This binary is designed to address the digitisation of the HCAL ring (CalibrHCALOther). This script is run on the root file outputs for the single fixed long-lived neutral kaon energy.

Inputs:

1. Input root files, wildcards can be used.
2. True energy of kaonL being simulated.
3. Output path to send the calibration text file and the output plot to.

This script joins together all the histograms (from all input root files) of direction corrections applied to the cells within the HCal in the barrel, endcap and other (other is

ring) for the long-lived neutral kaon samples. There is no iteration required for this step. The mean direction corrections for each of the three distributions (barrel, endcap, other) should be recorded and saved for a later stage.

3.4 SimCaloHitEnergyDistribution

This binary has two functions. The first is to continue the digitisation constant calibration of the HCal other and the second is to determine the MIP scale in both the ECal and the HCal if you wish to use the realistic digitisation options in ILDCaloDigi. This script is run on the root file outputs for the single muon events.

Inputs:

1. Input root files, wildcards can be used.
2. True energy of kaonL being simulated.
3. Output path to send the calibration text file and the output plot to.

The output once this executable has been run is a series of plots containing the direction corrected SimCalorimeterHit energies for the ECal (as a whole) and the HCal (Barrel, EndCap and Other separately). These plots should show a clear MIP peak in all cases.

CalibrHCALOther is set to be equal to CalibrHCALEndcap multiplied by the ratio of the MIP peaks for the EndCap and Other (from SimCaloHitEnergyDistribution), multiplied by the ratio of the absorber scintillator ratio (from gear files) and multiplied by the ratio of direction corrections (from HCalDigitisation_DirectionCorrectionDistribution)

$$\begin{aligned}
 \text{CalibrHCALOther} &= \text{CalibrHCALEndcap} \\
 &\times \frac{\langle \text{Direction Correction}_{\text{EndCap}} \rangle}{\langle \text{Direction Correction}_{\text{Ring}} \rangle} \\
 &\times \frac{\text{MIP Peak}_{\text{EndCap}}}{\text{MIP Peak}_{\text{Ring}}} \\
 &\times \frac{\text{Absorber Thickness}_{\text{EndCap}}}{\text{Absorber Thickness}_{\text{Ring}}} \\
 &\times \frac{\text{Scintillator Thickness}_{\text{Ring}}}{\text{Scintillator Thickness}_{\text{EndCap}}}
 \end{aligned} \tag{3}$$

The realistic digitisation variables are set as follows, CalibHCALMIP is set to be the peak value in the direction corrected SimCalorimeterHit energy distribution for the HCal barrel and CalibECALMIP is set to be the peak value in the direction corrected SimCalorimeterHit energy distribution for the ECal. Note: There is some ambiguity about

choosing the MIP scale in the HCal based entirely on the Barrel, but from experience the peaks in the distributions are very similar for both the Barrel and EndCap so the choice doesn't matter. If these differ significantly in your studies then you should reevaluate this procedure.

No iteration is required at this stage.

4 PandoraPFACalibration

4.1 PandoraPFACalibration_HadronicScale_MipResponse

This binary calibrates the MIP response in PandoraPFA. This script is run on the root file outputs for the single muon samples.

Inputs:

1. Input root files, wildcards can be used.
2. True energy of muons being simulated.
3. Output path to send the calibration text file and the output plot to.

This script makes plots of the direction corrected calorimeter hit energy for the HCal, ECal and Muon Chamber. The constants ECalToGeV MIP, HCalToGeV MIP and MuonToGeV MIP are then defined as the constants, which are needed to shift the position of the observed peak in these plots to 1. This determines the response of each part of the detector to MIP like particles and can be used to convert energy measurements into a number of normally incident mips required to deposit the same amount of energy. There is no iteration applied here.

4.2 PandoraPFACalibration_EMScale

This script is run on the root file outputs for the single photon samples and aims to set ECal/HCalToEMGeVCalibration.

Inputs:

1. Full path to photon root files. Wildcards can be used here.
2. True energy of photon.
3. Fractional calibration accuracy of PandoraPFA step. This is used to define the bin width in the plot, which determines precision of results. Default: 0.5%
4. Output path to send the calibration text file and the output plot to.

- Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.

After running this script on the root files you rescale ECalToEMGeVCalibration by:

$$\frac{\text{True Photon Energy}}{\text{Mean Reconstructed PFO Energy}} \quad (4)$$

Mean Reconstructed PFO Energy is generated by the binary once it is run over the root files. Iteration is then performed until the PFO energy for photon events falls within the desired accuracy. Iteration involves changing the ECalToEMGeVCalibration value in your Marlin steering files and then rerunning the photons through Marlin. Iteration keeps occurring until the the PFO energy for photon events is within acceptable limits defined by the fractional accuracy of the PandoraPFA stage. Once this stage is completed set HCalToEMGeVCalibration to equal ECalToEMGeVCalibration.

4.3 PandoraPFA Calibration_HadronicScale_ChiSquaredMethod

This script is run on the root file outputs for the single fixed kaonL energy and aims to set ECal/HCalToHadGeVCalibration.

Inputs:

- Full path to kaonL root files. Wildcards can be used here.
- True energy of kaonL particles being simulated.
- Fractional calibration accuracy of PandoraPFA step. This is used to define the bin width in the plot, which determines precision of results. Default: 0.5%
- Output path to send the calibration text file and the output plot to.
- Number of HCal layers for the detector being simulated. This is the number of layers in the HCal being simulated. This is needed to define a contained event.

This binary forms the 2D histogram of ECalToHad energy vs HCalToHad energy. A line of best fit is applied to the data set, once a cut is applied to remove leakage effects and the intercepts of this best fit are used to rescale ECal/HCalToHadGeVCalibration until both intercepts match the available energy of the kaonL being simulated. Again this requires iteration.

ECalToHadGeVCalibration and HCalToHadGeVCalibration are independently rescaled respectively by:

$$\frac{\text{True KaonL Energy}}{\text{ECalToHad PFO Energy Intercept}} \text{ and } \frac{\text{True KaonL Energy}}{\text{HCalToHad PFO Energy Intercept}} \quad (5)$$

The ECalToHad and HCalToHad PFO energy intercepts are generated by the binary. Iteration involves rescaling ECalToHadGeVCalibration and HCalToHadGeVCalibration and then rerunning the kaonL samples through Marlin again. The calibration constants are then rescaled again and this continues until the ECalToHad and HCalToHad PFO energy intercepts fall within the desired accuracy targeted from the PandoraPFA stage.

4.4 PandoraPFACalibration_HadronicScale_TotalEnergyMethod

This script is run on the root file outputs for the single fixed kaonL energy and is an alternative method for setting ECal/HCalToHadGeVCalibration.

Inputs:

1. Full path to kaonL root files. Wildcards can be used here.
2. True energy of kaonL particles being simulated.
3. Fractional calibration accuracy of PandoraPFA step. This is used to define the bin width in the plot, which determines precision of results. Default: 0.5%
4. Output path to send the calibration text file and the output plot to.
5. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.
6. Number of HCal layers for simulated detector. This is the number of layers in the HCal being simulated. This is needed to define a contained event.

This binary performs a Gaussian fit to total PFO energy, once both ECal/HCalToHad have been independently rescaled. The mean of this Gaussian fit is fixed and the ideal rescaling constants (one for ECalToHad one for HCalToHad) minimise the standard deviation of that Gaussian fit. The scale factors produced for the ideal fit, which are generated by the script, are used to rescale ECal/HCalToHadGeVCalibration independently.

Iteration involves rescaling ECalToHadGeVCalibration and HCalToHadGeVCalibration and then rerunning the kaonL samples through Marlin again. The calibration constants are then rescaled again and this continues until the rescaling factors produced by this method are both equal to unity within the errors defined to be the desired accuracy of the PandoraPFA stage.

4.5 PandoraPFACalibration_HadronicEnergyGaussianFit

This script is run on the root file outputs for the single fixed kaonL energy. It is used for applying Gaussian fits to reconstructed kaonL PFO energy, for events contained within the HCal.

Such events can also be used for the implementation of non-linearity corrections designed to force a linear relationship between simulated and reconstructed kaon samples across a range of different simulated energies, however, by default it is not recommended that you used these non-linearity corrections in your simulation.

Inputs:

1. Full path to kaonL root files. Wildcards can be used here.
2. True energy of kaonL.
3. Fractional calibration accuracy of PandoraPFA step. This is used to define the bin width in the plot, which determines precision of results. Default: 0.5%
4. Output path to send the calibration text file and the output plot to.
5. Percentage of data to perform Gaussian fit to. This range is taken to be the continuous range of data with the minimum root mean squared (RMS). The default is set to 90% of data.
6. Number of HCal layers for simulated detector. This is the number of layers in the HCal being simulated. This is needed to define a contained event.
7. Non linearity constants on or off (just sets the labels for the output).

This binary produces the Gaussian fit to and histogram of PFO energy for the kaonL data. A text file contains the mean, standard deviation and amplitude of the gaussian fit is also produced. It would be the mean and true energy used as inputs for non linearities if they were implemented.

4.6 PandoraPFACalibration_SoftwareCompensation

This script is run on root files produced by the software compensation training algorithm in the LCContent library. These root files have to be produced for different single kaon0L energies depending on the energy range of particles you will observe in your detector. For the ILC experiment, we recommend to use single kaon0L energies from 10 GeV to 100 GeV by step of 10 GeV. This script is used to re-calculate the software compensation weights used for hadronic energy corrections in the LCContent library.

Inputs:

1. The kaon0L root file pattern. For example, if your files look like SCTraining_10GeV.root, SCTraining_20GeV.root, ... then you must specify SCTraining_%.{energy}GeV.root. The pattern %.{energy} will be replaced by the specified energies (see next input)

2. The input kaon0L energies. It must be provided as a single string join with the character ':'. For example if you use energies from 10 GeV to 100 GeV by step of 10 GeV, you must pass the string 10:20:30:40:50:60:70:80:90:100. These energies will replace the pattern in the `%{energy}` of the previous input.
3. Output path to send the calibration text file and the output plot to.
4. The tree name in kaon0L root files.
5. Whether you want to use the raw cluster energy instead of the true kaon0L energy in the χ^2 computation. This option has been introduced for testing purpose only, do not use it.

This binary uses the Minuit (Migrad) minimizer to minimize a χ^2 and re-optimize the software 9 compensation weights. The text file output contains the new re-optimized weights and information on user inputs. The obtained weights may be replaced in the Pandora settings xml file in the xml element provided for that purpose :

```
<SoftwareCompensation>
  <SoftwareCompensationWeights>
    ... put your weights here ...
  </SoftwareCompensationWeights>
</SoftwareCompensation>
```

5 Suggested Running Order of Calibration

It is important to ensure that this calibration procedure is applied in the correct order. As changes to each variable will effect the simulation downstream of it necessary to calibrate the simulation in such a way that mirrors the reconstruction chain. Therefore, the digitisation stage will be calibrated before the PandoraPFA stage. In both cases it is also essential to set the MIP scale first, as both the digitiser (realistic options only) and PandoraPFA apply cuts based in units of MIPs. Once the MIP scale is set you can then begin the calibration procedure of the variables relating to the processor in question. Based on that logic here is a suggested calibration chain that you should use if you wish to use the realistic digitisation options in the ILDCaloDigi processor:

1. SimCaloHitEnergyDistribution - Set the MIP scale in the digitiser
2. ECalDigitisation_ContainedEvents - Digitsation for ECal
3. HCalDigitisation_ContainedEvents - Digitsation HCal Barrel and HCal EndCap simultaneously.

4. HCalDigitisation_DirectionCorrectionDistribution and SimCaloHitEnergyDistribution
- Get variables and set digitisation of HCal other
5. PandoraPFACalibrate_MipResponse - MIP scale in Pandora
6. PandoraPFACalibrate_EMScale
7. PandoraPFACalibrate_HadronicScale_ChiSquareMethod or PandoraPFACalibrate_HadronicScale_TotalE

If you do not require the realistic options in the ILDCaloDigi processor the procedure is the same, just omitting step 1.